Eastern Region Training and Applications Note No. 12

ER TAN-12

ESTABLISHING A WORLD WIDE WEB SERVER ON THE SCIENTIFIC APPLICATIONS COMPUTER (SAC)

JONATHAN BLAES

National Weather Service Forecast Office

Albany, New York

Scientific Services Division

Eastern Region Headquarters

Bohemia, New York

March, 1997

1. INTRODUCTION

This paper will provide a guide for those in the National Weather Service (NWS) who wish to establish a World Wide Web (WEB) server on the Scientific Applications Computer (SAC). The paper outlines how to connect the SAC to the Internet, how to obtain free WEB server software and how to install it. The SAC is the cornerstone of the SOO/SAC Program, a NWS program providing the Science and Operations Officer (SOO) at each NWS Weather Forecast Office with tools for on-station research, training, and professional development (Bruehl 1996). Since some technical information is presented, a basic knowledge of computer skills, especially in the UNIX environment, is assumed. It is also assumed that the reader has a general knowledge of the Internet and the World Wide Web. However, some references and resources are provided to supply additional information.

The WEB encompasses a vast amount of real time, research, reference and general meteorological information, as well as enormous amounts of information on nearly any other subject. The World Wide Web offers tremendous potential for research and professional development. In addition, operating a WEB server provides a relatively easy way to share and display large amounts of meteorological and other types of data.

2. GETTING CONNECTED

There are numerous ways to connect to the Internet. Probably the most logical way is through a direct connection. This involves having a dedicated, high capacity line that connects your computer to the Internet. If you have direct access to the Internet through an educational institution or commercial service, the process is quite simple. All that it is involved is plugging in the hardware, connecting the line and configuring your system.

Connecting your SAC directly to the Internet provides the potential to have your own WEB server. The WEB server is a program that enables you to distribute WEB pages directly off your computer. Without a direct Internet connection and an associated WEB server, you will need to have your WEB pages distributed by another machine. Most NWS users will have their SAC already connected to the Internet, but if you do not, Peggy Bruehl, the NWS SOO/SAC coordinator, who maintains the SOO/SAC Home Page has some instructions available at: http://www.comet.ucar.edu/pub_html/sac_html/index.html(see Bruehl 1995).

3. SETTING UP A WEB SERVER

Installation of a WEB server allows the maximum potential for the development of applications associated with WEB documents, including the use of optional features such as Common Gateway Interfaces (CGI) to support interactive documents and other application programs. To run your own WEB server you will need to have direct Internet access on the machine in which you intend to run the server. Servers can run on most types of computer platforms. NCSA, the National Center for Supercomputing Applications, has compiled and uncompiled versions of their WEB server, HTTPd, available as freeware. The following pages contain information on the installation, compilation, configuration, startup, shutdown, and testing of the NCSA WEB server.

The instructions and guidelines are based on information from the NCSA HTTPd Home Page (see NCSA 1995a), The WEB Server Book (see Magid et al. 1995) and Managing Internet Information Services (see Liu et al. 1994).

The current version of NCSA's Web server is HTTPd_1.5.2. The SAC is a Hewlett Packard model HP 715/64 that operates on the HP-UX 9.03 UNIX operating system and NCSA has an uncompiled version of HTTPd available for this machine.

a. Compilation of NCSA HTTPd Server Software

```
[Note : In the following, lines that are bold and begin with % are
to be inputted by the user.]

     The NCSA HTTPd source code is available from the following
WEB address and by anonymous ftp, respectively:

     http://hoohoo.ncsa.uiuc.edu/docs/setup/compilation.html
```

ftp://ftp.ncsa.uiuc.edu/Web/httpd/UNIX/ncsa_httpd/current/httpd_source.tar.Z

To begin the compilation process, login as root. Before getting the source code, you need to create the /usr/local/src directory.

```
% mkdir /usr/local/src
```

Change to the /usr/local/src directory.

```
% cd /usr/local/src
```

Get the source code via the WEB or FTP and save it as ncsa_source.tar.Z in the /usr/local/src directory.

The following command will unpack the code.

```
% zcat ncsa_source.tar.Z | tar xvf -
```

The unpacking process will create a subdirectory called httpd_1.5.2 that contains a README file, the Makefile, and several subdirectories. Subdirectories include cgi-bin, conf, icons, logs, src, and support.

To build the HTTPd server binary, change directories to the source directory and make a backup copy of the Makefile using the following commands:

```
%  cd httpd_1.5.2/src

%  cp Makefile Makefile.dist
```

Now edit the original Makefile while keeping in mind not to change any variables unless you know exactly what they do. The Makefile can be edited in any text editor; for the SAC you can use the text editor associated with the edit button on the VUE panel.

Define which compiler is used on your machine by uncommenting the Makefile line referring to your particular compiler. The SAC has a cc compiler. Select the cc compiler by removing the # in front of the cc line while ensuring that a # is in front of the other lines that refer to different compilers.

You are now ready to compile the server. Type in the following command:

```
%  make hp-cc
```

This will compile the source code. After compilation is over, the HTTPd binary needs to be copied into its permanent directory.

Copy the HTTPd binary into the /usr/local/etc/httpd directory.

```
%  cp  /usr/local/etc/httpd/src/httpd /usr/local/etc/httpd/
```

b. HTTPd Configuration

There are three configuration files located in the /usr/local/etc/httpd/conf directory which must be modified.

The three configuration files are:

httpd.conf	The main server configuration file
srm.conf	The server resource configuration file
access.conf	The global access control file

These files are originally found with a .conf-dist file extension. Make copies of these with a new extension of .conf, using the .conf-dist as originals for safe keeping and reference.

```
% cp httpd.conf-dist  httpd.conf

% cp srm.conf-dist  srm.conf

% cp access.conf-dist  access.conf
```

1) Standard httpd.conf setup file

The httpd.conf controls most of the basic and technical elements of server operation. For general public access to the server the httpd.conf file should be configured with the following settings. Under most circumstances the default value for the settings will be fine, but there are a few settings which need to be adjusted. A brief summary of the more important settings are listed below.

ServerType: The default setting of standalone should be used under most circumstances.

ServerRoot: This setting is the direct path to the directory where you put the httpd binary. It also determines where the files that will be served are located. The default is /usr/local/etc/httpd/

ServerName: The official hostname of your server as it appears in the URL.

ServerAdmin: The E-mail address given to users to report problems or comments. This provides a consistent address for your site even if the administrator changes. It is usually of the form Webmaster@your.domain.

User and Group: This defines the user id (UID) and the group id (GID) that the server uses. These settings are often a problem for those first beginning WEB administration. It may take a little adjusting to do this correctly. The default UID is nobody and the GID is -1. If there is a problem, trial and error will work best here.

Port: The network port on which the server is running. The default is 80.

StartServer: The number of servers launched into the pool of operating servers when HTTPd is started. The default value is fine under most circumstances.

MaxServer: The maximum number of servers available

in the server pool. The default value is fine under most circumstances.

PidFile: The name of the file where the original server logs its own process id number. The default setting is fine and this will send the data to /logs/httpd.pid.

ErrorLog: The default path to /logs/error_log is fine.

Agentlog: The default path to /logs/agent_log is fine.

RefererLog: The default path to /logs/referer_log is fine.

Look over the setup one more time and then save it.

2) Standard srm.conf setup file

The srm.conf file controls how the HTTPd serves files. Under most circumstances the default values for the settings will be acceptable. The following is a general description of two of the more common settings.

DocumentRoot: This setting specifies the directory from which the files, images or other documents are served from. The default setting is /usr/local/etc/httpd/htdocs. The htdocs directory may need to be created if it was not done so during the unpacking process earlier. This can be done with the following command:

```
               % mkdir /usr/local/etc/httpd/htdocs
```

Files or documents outside of this directory can still be served with the use of a symbolic link. A symbolic link is a Pointer to another directory or file. To set up a symbolic link use the following command:

```
% ln -s /the path to where you want files served from/
```

/usr/local/etc/httpd/htdocs/

UserDir: This setting specifies which directory users on the system can use to serve files to the public from their home directories. The default value is public_html which allows users on the system to serve files in Their own public_html directories. To prohibit this possible security risk, set UserDir to DISABLED.

The remaining settings or directories in the srm.conf configuration file will only need to be Adjusted if you implement advanced or technical features on your WEB server.

3) Standard access.conf setup file

The access.conf controls certain security issues, access to certain directories, and the type of access WEB browsers have to your WEB server. For general, public access to the server you should configure the access.conf file with the following settings. Under most occasions the default value for the settings should do, but here is a general description of the more common settings.

AllowOveride: Change this setting from All to None, This enables another layer of security.

Options: Change the options setting from Indexes to FollowSymLinks. This directive prohibits users from randomly looking through your directory.

Directory: This is the second directory directive. It should be changed to whatever directory you have set the document root to.

c. HTTPd Server Startup

The server is easily started from the command Line by executing the binary with the following command:

```
%  /usr/local/etc/httpd/httpd &
```

The server will need to be started every time the SAC is rebooted. To get the server to start on its own every time the machine is booted up, you will need to edit the inittab file. Add the following line to the inittab file which is usually located in the etc directory.

```
home::bootwait:/usr/local/etc/http/httpd &
```

d. HTTPd Server Testing

To test the server and see that it is running, type in the following:

```
%  telnet localhost 80
```

Where the 80 above is the network port server your server is operating on. You will see the following:

```
trying...localhost aaa.aaa.aaa

Escape character is ctrl]
```

Where aaa.aaa.aaa is the host domain name of your address. This opens up a connection to your server. To test the server, type in the following and hit return rapidly twice.

```
%  HEAD/HTTP/1.0
```

You will see the following status message:

```
HTTP 1.5 OK
```

e. HTTPd Server Termination

To kill the server identify the process id number and then kill it.

```
%  ps -ef

%  kill [process id number]
```

Or change directories and kill the log file.

```
%  cd /usr/local/etc/httpd

%  kill 'cat /logs/http.pid'
```

f. WEB Server Security

Security is an extremely important issue on the Internet. If you are not directly connected to the Internet and only access it through dial-in capability then there is little threat from the outside. However, if your system is directly connected to the Internet then there is a security risk. The first thing to keep in mind is that it is very important to have a good security setup on the host system. Once you are confident of this, then you need to look at the setup of the server itself.

When configuring your server there are several things you can do to improve the security. The two biggest directives to change for security reasons are in the access.conf setup file. Change the AllowOveride directive to None and change the Options directive to FollowSymLinks (NCSA 1995b).

For more advanced security concerns there are many more options. Security at the advanced level can be quite a complex subject and to ensure the brevity of this document, a full discussion is not included. However, two popular methods for serious security concerns are worth mentioning. The first method, firewalls, are a special type of gateway host which acts as a wall between your network and the

Internet. The firewall is setup to prohibit most traffic from going into your system and allowing data to go out. More information on firewalls and firewall tool kits are available on-line at http://www.zeuros.co.uk/firewall. Another popular method is the use of wrapper programs, particularly the program TCP Wrappers. These programs act as a wrapper around the server program. The software allows you to setup the program to accept or deny any connection based on various criteria while maintaining a log of all attempted network connections. More information on TCP Wrappers is available at ftp://ftp.win.tue.nl/pub/security.

4. CREATING A WEB PAGE

The HyperText Markup Language (HTML) is composed of a set of elements that simply enhance and determine the display of a simple text document. There are numerous versions of HTML, each new version is usually just an expanded version of an older one. Different browsers and different versions of a particular browser can read different versions of HTML. Nearly all browsers can read the earliest versions of HTML. Numerous descriptions of the most commonly used HTML elements as well as new elements and features are available on-line. Three recommended on-line resources include HTML Quick Reference (see Grobe 1995), HTML Elements List (see Hannah 1995) and A Beginner's Guide to HTML (see NCSA 1996). The Appendix lists some of the most commom HTML elements.

Remember that the World Wide Web (WEB) is a network of computers that supply WEB documents, data, multimedia information and much more. The Home Page is the first place or first document you see when you visit a particular WEB site. The Home Page usually provides a greeting, an introduction, and an index of all the information available at that particular site. It is important to keep in mind that HTML is an evolving language, and different WEB browsers may recognize slightly different sets of HTML elements.

5. SUMMARY

This training and applications note contains the basic information needed to establish a presence on the World Wide Web for those members of the NWS community who have access to a SAC. Instructions and references have been presented discussing how to connect the SAC to the Internet, obtain free WEB server software, install the server software, and use the basic techniques of WEB page creation. The WEB offers a relatively reliable, cost effective, and easy means to share and receive large quantities of meteorological information. The WEB contains a vast amount of material on nearly any other subject as well. Operating a

WEB server allows you to pursue opportunities in community outreach, public relations, research, and professional development. The possibilities are unlimited.

REFERENCES

Bruehl, P., National Weather Service-Office of Meteorology, cited 1995: Connecting your SAC to the Internet. [Available on-line from http://www.comet.ucar.edu/pub_html/sac_html/ README/internet.html.]

_____, P., National Weather Service-Office of Meteorology, cited 1996: SOO/SAC Home Page. [Available on-line from http://www.comet.ucar.edu/pub_html/sac_html/index.html.]

Grobe, M., Academic Computing Services, The University of Kansas, cited 1995: HTML Quick Reference. [Available on-line from http://kuhttp.cc.ukans.edu/lynx_help/HTML_quick.html.]

Hannah, M J, Sandia National Laboratories, cited 1995: HTML Elements List [Available on-line from http://www.sandia.gov/sci_compute/elements.html.]

Liu, C., J. Peek, R. Jones, B. Buus, and A. Nye, 1994: Managing Internet Information Services. O'Reilly & Associates, INC., 630 pp.

Magid, J., R. D. Mathews, and P. Jones, 1995: The Web Server Book. Ventana Press, 315 pp.

NCSA, cited 1995a: The NCSA HTTPd Home Page. [Available on-line from http://hoohoo.ncsa.uiuc.edu/.]

_____, cited 1995b: Making Your Setup More Secure. [Available on-line from http://hoohoo.ncsa.uiuc.edu/docs/tutorials/security.html.]

_____, cited 1996: A Beginner's Guide to HTML. [Available on-line from http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html.]

APPENDIX

1. INTRODUCTION TO HTML DOCUMENTS

The following section shows the basic elements needed to create an HTML document. A sample HTML document and its results are included.

a. Elements placed in the body

Text Elements

`<p>...</p>`
This element marks the beginning and end of a paragraph that will be formatted before it is displayed on the screen. This allows a paragraph to keep its continuity on the screen by eliminating line breaks.

`<pre>...</pre>`
Identifies text that has already been preformatted in some format and must be displayed with these characteristics. Such text may include prearranged columns or tables.

`<blockquote>...</blockquote>`
Includes a section of text quoted from some other source.

Hyperlinks or Anchors

A hyperlink is an element that links the current WEB document to other WEB documents. An anchor must include a name or href attribute, and may include both. There are several optional attributes, but they are rarely encountered.

`<a href="URL">...</a>`
Link to another file or WEB resource.

`<a name="anchor_name">...</a>`
Define a target location in a WEB document.

`<a href="#anchor_name">...</a>`
Link to a target location in the same WEB document.

Headers
`<h1>...</h1>` Largest Font
`<h2>...</h2>`

<h3> . . . </h3>
<h4> . . . </h4>
<h5> . . . </h5>
<h6> . . . </h6> Smallest Font

Physical Styles
<b> . . . </b> Boldface
<I> . . . </I> Italics
<u> . . . </u> Underline
<ul>. . . </ul> Present an unordered list

```
<ul>
<li>An item in the list
</ul>
```

<ol>. . . </ol> Present an ordered list

```
<ol>
<li>An item in the list
</ol>
```

b. An Example HTML Document

The following set of code is a portion of the HTML document that contains information for the Skywarn Spotter Organization at the NWSFO Albany, NY. This portion of the code contains many different HTML elements such as images, font changes, ordered lists, line breaks, as well as hyperlinks to meteorological images and text from other WEB servers. Figure A1 represents a screen snapshot of the browser's interpretation of the code.

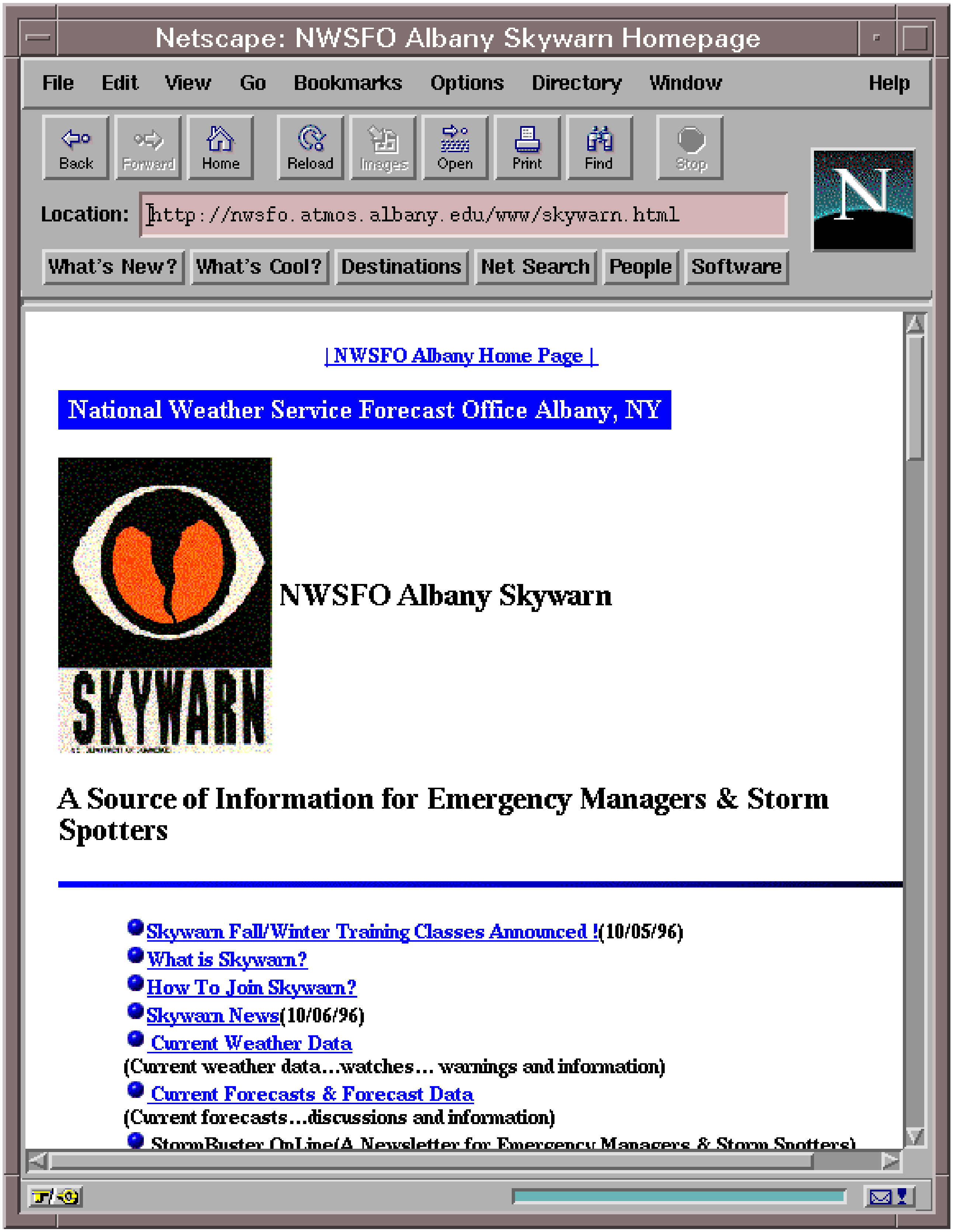
```
   <html>

   <head>

   <title>
NWSFO Albany Skywarn Homepage

   </title>

   <body alink="#FF0000" vlink="#3232CD" link="#0000FF"
bgcolor="#FFFFFF" text="#000000"
```

```
    <body>

    <center>

    <h4>

    <A HREF=http://nwsfo.atmos.albany.edu/>| NWSFO Albany Home
Page |

    </a>

    </center>

    </h4>

    <img src="top2.gif">

    <H2>

    <IMG SRC="skywarn2.gif" align=middle> NWSFO Albany Skywarn

    <br>

    <br>
A Source of Information for Emergency Managers & Storm Spotters

    </H2>

    <IMG Src="line.marble1.gif">

    <h4>

    <ul>

    <IMG Src="abball.gif">

    <a href=skywrncls2.html>Skywarn Fall/Winter Training Classes
Announced !

    </a>(10/05/96)

    <br>

    <IMG Src="abball.gif">

    <a href="#whatsky">What is Skywarn?

    </a>
```

```
   <br>

   <IMG Src="abball.gif">

   <a href="#join">How To Join Skywarn?

   </a>

   <br>

   <IMG Src="abball.gif">

   <a href="#skynews">Skywarn News

   </a>(10/06/96)

   <br>

   <IMG Src="abball.gif">

   <a href="#Current">
Current Weather Data

   </a>

   <br>
(Current weather data...watches...warnings and information)

   <br>

   <IMG Src="abball.gif">

   <a href="#fcst">
Current Forecasts & Forecast Data

   </a>

   <br>
Current forecasts...discussions and information)

   <br>

   <IMG Src="abball.gif">
StormBuster OnLine(A Newsletter for Emergency Managers & Storm
Spotters)

   <br>

   </body>
```

```
</html>
```

Figure A1. Screen snapshot of the Netscape browser interpretation of code.


Netscape: NWSFO Albany Skywarn Homepage
File Edit View Go Bookmarks Options Directory Window Help
Back Forward Home Reload Images Open Print Find Stop
Location: http://nwsfo.atmos.albany.edu/www/skywarn.html
What's New? What's Cool? Destinations Net Search People Software

| NWSFO Albany Home Page |

National Weather Service Forecast Office Albany, NY

SKYWARN

NWSFO Albany Skywarn

A Source of Information for Emergency Managers & Storm Spotters

- Skywarn Fall/Winter Training Classes Announced !(10/05/96)
- What is Skywarn?
- How To Join Skywarn?
- Skywarn News(10/06/96)
- Current Weather Data
(Current weather data...watches... warnings and information)
- Current Forecasts & Forecast Data
(Current forecasts...discussions and information)
- StormBuster OnLine(A Newsletter for Emergency Managers & Storm Spotters)

Figure A1